Programación I

Grado de Ingeniería Informática y Doble Grado II-ADE - ETSIINF - UPM

Examen Eval2

18/12/2018

Realización: El ejercicio se realizará en la hoja de respuestas, que será lo único que se entregará. En ella se harán constar los apellidos y el nombre. Se pueden utilizar hojas de sucio aparte. Las hojas de sucio **NO** se entregan.
Duración: El tiempo para realizar este examen es de **1 hora y media.**
Calificaciones: Las calificaciones se publicarán el **8 de Enero.**

En España, el NIF o Número de Identificación Fiscal consta de ocho dígitos y una letra mayúscula (no Ñ) al final.

Ejercicio 1 (4 puntos)

Se pide implementar en Java una función **tieneFormatoNIF** que recibe un array de caracteres y devuelve `true` si los elementos del array tienen el formato de un NIF (ocho dígitos y una letra mayúscula, no Ñ, al final) y `false` en caso contrario.

Ejemplos:

```
tieneFormatoNIF(['0','3','4','4','5','0','8','5','Z']) → true
tieneFormatoNIF(['0','3','4','?','5','0','8','5','Z']) → false
tieneFormatoNIF(['0','3','4','9','5']) → false
```

Se pueden usar las siguientes funciones:

```
static boolean esLetraMayuscula (char caracter){
   return 'A' <= caracter && caracter <= 'Z';
}

static boolean esDigito (char caracter){
   return '0' <= caracter && caracter <= '9';
}
```

Ejercicio 2 (4 puntos)

Se pide implementar en Java una función **aNum** que, dado un array de caracteres que tiene el formato de un NIF, devuelve el número entero(*) correspondiente al NIF sin la letra. La precondición es que el array que recibe **aNum** tiene el formato de un NIF (ocho dígitos seguidos de una letra mayúscula que no es la Ñ).

Ejemplos:

```
aNum(['0','8','9','2','8','8','6','7','Z']) → 8928867
aNum(['0','0','0','0','0','0','2','4','R']) → 24
```

(*) Nota: la función devuelve un número entero, no un String.

Se puede usar la siguiente función:

```
static int aDigito (char caracter){
   return caracter - '0';
}
```

Ejercicio 3 (2 puntos)

La letra que aparece en la última posición del NIF es un código de control que sirve para comprobar si un NIF es correcto o se trata de una falsificación. Para comprobar si un NIF es correcto se calcula el número del NIF módulo 23 (el resto de dividirlo por 23) y se comprueba que el valor del módulo se corresponde con una letra según la siguiente tabla:

Resto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Código	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Por ejemplo, al número 08928867 le corresponde la letra Z ya que 8928867 módulo 23 es 14.

Se pide:

Implementar en Java una función **esNIFCorrecto** que, dado un array de caracteres de cualquier tamaño y contenido, devuelve `true` si dicho array representa un NIF correcto y `false` en caso contrario. Se deberá usar la función **letraNIF** que dado un número entre 0 y 22 devuelve la letra mayúscula correspondiente según la tabla.

Ejemplos:

```
esNIFCorrecto(['0','8','9','2','8','8','6','7','Z']) → true
esNIFCorrecto(['0','0','0','0','0','0','2','4','R']) → true
esNIFCorrecto(['0','0','0','0','0','0','2','4','B']) → false
esNIFCorrecto(['0','3','4','9','5']) → false
```

Programación I - Grado de Ingeniería Informática y Doble Grado II-ADE - ETSIINF - UPM

Examen Eval2 - 18/12/2018 - **Hoja De Respuestas**

Apellidos:_____Nombre:_____

DNI:_____ Marque si cursa el doble Grado II+ADE ☐

Ejercicio 1

```
static boolean tieneFormatoNIF (char[] arr){

  // Si el array no tiene nueve elementos o el último no es una letra mayúscula
  // entonces el array no tiene el formato de un NIF
  if (arr.length != 9 || !esLetraMayuscula(arr[arr.length-1])){
    return false;
  }
  // el array tiene nueve elementos y el último es una letra mayúscula
  // falta ver si todos los elementos menos el último son un dígito
  boolean formatoNIF = true;
  int i = 0;

  // mientras queden elementos en el array (excepto el último) y tenga formato NIF
  while (i < arr.length - 1 && formatoNIF){

    if (!esDigito(arr[i])){
      // si el elemento no es un dígito entonces no está en formato NIF
      formatoNIF = false;
    }
    else{
      // si el elemento es un dígito entonces se pasa al siguiente elemento
      i++;
    }
  }//del while
  return formatoNIF;
}
```

Ejercicio 2

```
static int aNum (char[] nif){
  // se declara una variable donde se va a almacenar el resultado
  int num = 0;

  // se recorre el array de digitos
  for (int i = 0; i < 8; i++){
    // se va almacenando el resultado
    num = num * 10 + aDigito(nif[i]);
  }
  return num;
}
```

Ejercicio 3

```
static boolean esNIFCorrecto (char[] arr){
 return tieneFormatoNIF(arr) && letraNIF(aNum(arr) % 23) == arr[8];
}
```